STOCK PRICE PREDICTION USING LSTM

A SOFTWARE DESIGN PROJECT- III REPORT

Submitted by

V VAISHNAV	**(17113027)**
R V SIDHARTH	**(17113035)**
SUDHARSHAN U	**(17113059)**

Under the guidance of

Dr. S.SATHYALAKSHMI

in partial fulfillment for the award of the degree

of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING

HINDUSTAN
INSTITUTE OF TECHNOLOGY & SCIENCE
(DEEMED TO BE UNIVERSITY)

NOVEMBER 2020

HINDUSTAN
INSTITUTE OF TECHNOLOGY & SCIENCE
(DEEMED TO BE UNIVERSITY)

BONAFIDE CERTIFICATE

Certified that this project report **STOCK PRICE PREDICTION USING LSTM** is the bonafide work of **V Vaishnav (17113027), Sidharth R V (17113035), Sudharshan U (17113059)** who carried out the Software Design Project under my supervision during the academic year **2020-2021**

SIGNATURE

SUPERVISOR
Dr. S.SATHYALAKSHMI
PROFESSOR
DEPARTMENT OF CSE

INTERNAL EXAMINER | **EXTERNAL EXAMINER**

Name: ________________________ | Name: ________________________

Designation: _____________________ | Designation: _____________________

ACKNOWLEDGEMENT

We thank **Dr. Angelina Geetha**, Head of the Department of Computer Science & Engineering for her strong support and encouragement throughout our course of study.

We express deep gratitude to our supervisor Assistant Professor **Dr. S. Sathyalakshmi**, Department of Computer Science & Engineering for constant guidance and continued assistance in completion of this project

We thank all the **Faculty members and Technical staff** of Department of Computer Science and engineering for their valuable support and suggestions at various stages of design project development.

We are extremely indebted to our family members for their adorable support and care.

V. VAISHNAV

SIDHARTH R. V.

SUDHARSHAN U.

ABSTRACT

The prediction of a stock market web application that may serve as an early recommendation system for short-term investors and as an early financial distress warning system for long-term shareholders. Forecasting accuracy is the most important factor in selecting any forecasting methods. Research efforts in improving the accuracy of forecasting models are increasing since the last decade. The appropriate stock selections those are suitable for investment is a very difficult task. The key factor for each investor is to earn maximum profits on their investments. We used deep learning and more precisely one of the most famous recurrent neural networks : LSTM, in order to perform stock market prediction. Since it is necessary to mention that stock market prediction is not an easy task since the prediction part could be divided into two : fundamental analysis (sales, earning, profits, …) and technical analysis (historical price, …). Which means numerous factors could affect the stock price trends, but in this tutorial we used only time series forecasting using the historical price of a given stock. We used Long Short Term Memory, a commonly used RNN. Considering the type of data that we will be feeding our model and the ability of a RNN to allow information to persist unlike standard feed forward neural networks (they could only process single data points eg : images), an LSTM is the best fit for these type of problems. LSTM could easily process an entire sequence of data and it introduces the memory cell, which make the network able to effectively associate memories and input remote in time. In this example we fed our model with a set of sequences that will help predict a given price using time steps. The final predicted model will be displayed in a web-application, so this will be user-friendly.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 OVERVIEW

The stock market is a vast array of investors and traders who buy and sell stock, pushing the price up or down. The prices of stocks are governed by the principles of demand and supply, and the ultimate goal of buying shares is to make money by buying stocks in companies whose perceived value (i.e., share price) is expected to rise. Stock markets are closely linked with the world of economics —the rise and fall of share prices can be traced back to some Key Performance Indicators (KPI's). The five most commonly used KPI's are the opening stock price (`Open'), end-of-day price (`Close'), intraday low price (`Low'), intra-day peak price (`High'), and total volume of stocks traded during the day (`Volume'). Economics and stock prices are mainly reliant upon subjective perceptions about the stock market. It is near impossible to predict stock prices to the T, owing to the volatility of factors that play a major role in the movement of prices. However, it is possible to make an educated estimate of prices. Stock prices never vary in isolation: the movement of one tends to have an avalanche effect on several other stocks as well. This aspect of stock price movement can be used as an important tool to predict the prices of many stocks at once. Due to the sheer volume of money involved and number of transactions that take place every minute, there comes a trade-off between the accuracy and the volume of predictions made; as such, most stock prediction systems are implemented in a distributed, parallelized fashion. These are some of the considerations and challenges faced in stock market analysis.

1.2 OBJECTIVE

In the past decades, there is an increasing interest in predicting markets among economists, policymakers, academics and market makers. The objective of the proposed work is to study and improve the supervised learning algorithms to predict the stock price.

1.3 PROBLEM DEFINITION

Predicting stock market prices is a complex task that traditionally involves extensive human-computer interaction. Due to the correlated nature of stock prices, conventional batch processing methods cannot be utilized efficiently for stock market analysis. We propose an online learning algorithm that utilizes a kind of recurrent neural network (RNN) called Long Short Term Memory (LSTM), where the weights are adjusted for individual data points using stochastic gradient descent. This will provide more accurate results when compared to existing stock price prediction algorithms. The network is trained and evaluated for accuracy with various sizes of data, and the results are tabulated. A comparison with respect to accuracy is then performed against an Artificial Neural Network.

CHAPTER 2

ALGORITHM AND METHODOLOGY

2.1 GENERAL DEFINITION

Methodology is the systematic, theoretical analysis of the methods applied to a field of study. It comprises the theoretical analysis of the body of methods and principles associated with a branch of knowledge. Typically, it encompasses concepts such as paradigm, theoretical model, phases and quantitative or qualitative techniques. A methodology does not set out to provide solutions - it is therefore, not the same as a method. Instead, a methodology offers the theoretical underpinning for understanding which method, set of methods, or best practice's can be applied to a specific case, for example, to calculate a specific result.

2.2ALGORITHM

Input: Historical stock price data

Output: Prediction for stock prices based on stock price variation

1. Start
2. Stock data is taken and stored in a numpy array of 3 dimensions using MinMaxScalar.
3. The data is split into testing set and training set.
4. LSTM neural network structure is build.
5. Train the constructed network on the data
6. Use the output of the last layer as prediction of the next time step.
7. Repeat steps 5 and 6 until optimal convergence is reached.
8. Obtain predictions by providing test data as input to the network.

9. Evaluate accuracy by comparing predictions made with actual data.
10. End.

2.3 METHODOLOGY

Dataset is taken from the stock data of a particular company from the Symbol that is entered by the user. The data set contains information like previous closing, opening, high, low, volume of the stocks of that company. From these datasets, we extract only 65% of data; this data will be used to train the model. Using this trained set of data, predicting the next 10 days stock market price of that company can be accomplished. This closing price of stock is given preference as investors have to take decision on buying with only the stock closing value.

CHAPTER 3
SYSTEM ANALYSIS

3.1 EXISTING SYSTEM

Traditional approaches to stock market analysis and stock price prediction include fundamental analysis, which looks at a stock's past performance and the general credibility of the company itself, and statistical analysis, which is solely concerned with number crunching and identifying patterns in stock price variation. The latter is commonly achieved with the help of Genetic Algorithms (GA) or Artificial Neural Networks (ANN's), but these fail to capture correlation between stock prices in the form of long-term temporal dependencies. Another major issue with using simple ANNs for stock prediction is the phenomenon of exploding / vanishing gradient, where the weights of a large network either become too large or too small (respectively), drastically slowing their convergence to the optimal value. This is typically caused by two factors: weights are initialized randomly, and the weights closer to the end of the network also tend to change a lot more than those at the beginning. An alternative approach to stock market analysis is to reduce the dimensionality of the input data and apply feature selection algorithms to shortlist a core set of features (such as GDP, oil price, inflation rate, etc.) that have the greatest impact on stock prices or currency exchange rates across markets [10]. However, this method does not consider long-term trading strategies as it fails to take the entire history of trends into account; furthermore, there is no provision for outlier detection.

3.2 PROPOSED SYSTEM

We propose an online learning algorithm for predicting the end-of-day price of a given stock with the help of Long Short Term Memory (LSTM), a type of Recurrent Neural Network (RNN).

CHAPTER 4
SYSTEM DESIGN

4.1 LSTM – An Overview

LSTM's are a special subset of RNN's that can capture context-specific temporal dependencies for long periods of time. Each LSTM neuron is a memory cell that can store other information i.e., it maintains its own cell state. While neurons in normal RNN's merely take in their previous hidden state and the current input to output a new hidden state, an LSTM neuron also takes in its old cell state and outputs its new cell state. An LSTM memory cell, as depicted in Figure 4.1, has the following three components, or gates:

1. Forget gate: the forget gate decides when specific portions of the cell state are to be replaced with more recent information. It outputs values close to 1 for parts of the cell state that should be retained, and zero for values that should be neglected.

2. Input gate : based on the input (i.e., previous output o(t-1), input x(t), and previous cell state c(t-1)), this section of the network learns the conditions under which any information should be stored (or updated) in the cell state

3. Output gate: depending on the input and cell state, this portion decides what information is propagated forward (i.e., output o(t) and cell state c(t)) to the next node in the network.

Thus, LSTM networks are ideal for exploring how variation in one stock's price can affect the prices of several other stocks over a long period of time. They can also decide (in a dynamic fashion) for how long information about specific past trends in stock price movement needs to be retained in order to more accurately predict future trends in the variation of stock prices.

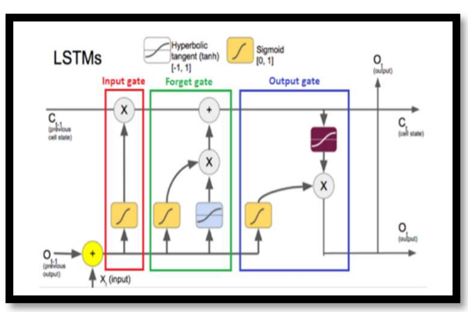

Fig – 4.1: An LSTM memory cell

4.2 OVERALL SYSTEM ARCHITECTURE

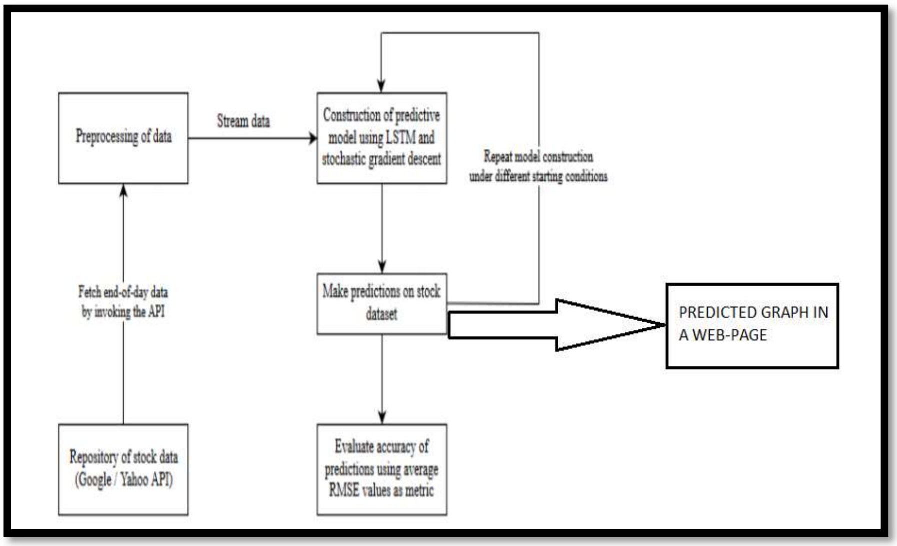

Fig – 4.2: LSTM-based stock price prediction system

The stock prediction system depicted in Figure 4.2 has three main components. A brief explanation of each is given below:

4.2.1 Obtaining dataset and pre-processing

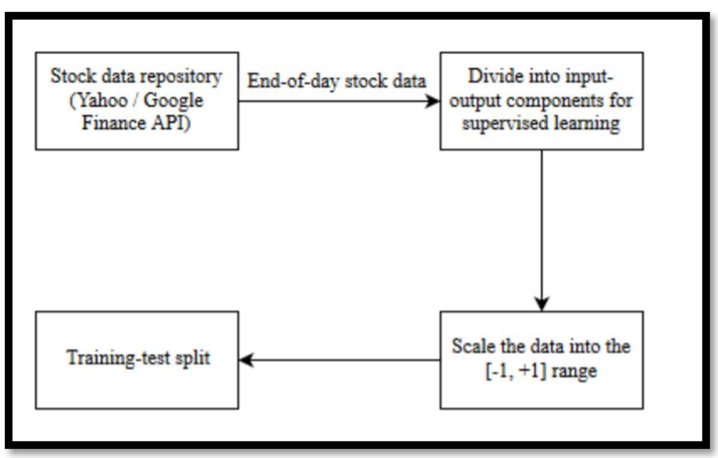

Fig – 4.2.1: Data pre-processing

Benchmark stock market data (for end-of-day prices of various ticker symbols i.e., companies) was obtained from two primary sources: Yahoo Finance and Google Finance. These two websites offer URL-based APIs from which historical stock data for various companies can be obtained for various companies by simply specifying some parameters in the URL. The obtained data contained five features:

1. Date: of the observation
2. Opening price: of the stock
3. High: highest intra-day price reached by the stock
4. Low: lowest intra-day price reached by the stock
5. Volume: number of shares or contracts bought and sold in the market during the day
6. OpenInt i.e., Open Interest: how many futures contracts are currently outstanding in the market

The above data was then transformed (Figure 4.2.1) into a format suitable for use with our prediction model by performing the following steps:

1. Transformation of time-series data into input-output components for supervised learning
2. Scaling the data to the [-1, +1] range.

4.2.2 Construction of prediction model

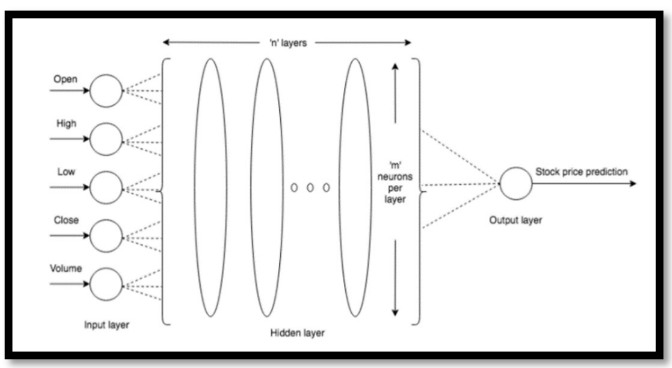

Fig – 4.2.2: Recurrent Neural Network structure for stock price prediction

The input data is split into training and test datasets; our LSTM model will be fit on the training dataset, and the accuracy of the fit will be evaluated on the test dataset. The LSTM network (Figure 4.2.2) is constructed with one input layer having five neurons, 'n' hidden layers (with 'm' LSTM memory cells per layer), and one output layer (with one neuron). After fitting the model on the training dataset, hyper-parameter tuning is done using the validation set to choose the optimal values of parameters such as the number of hidden layers 'n', number of neurons 'm' per hidden layer, batch size, etc.

CHAPTER 5

SOFTWARE AND HARDWARE REQUIREMENTS

5.1 SOFTWARE REQUIREMENTS

The application and the recognition package. Usually the recognition package is supplied as a DLL (Dynamic Link Library).

5.1.1 Python

Python is one of those rare languages which can claim to be both simple and powerful. You will be pleasantly surprised to see how easy it is to concentrate on the solution to the problem rather than on the syntax (i.e. the structure of the program that you are writing) of the language. Python is an easy to learn, powerful programming language. It has efficient high-level data structures and a simple but effective approach to object-oriented programming.

5.1.2 Python Libraries

The Python Standard Library is huge indeed. The proposed system uses Sklearn, Pandas, Numpy, Tensorflow, Keras, and many more libraries for the prediction.

5.1.3 Tensorflow

TensorFlow is an end-to-end open source platform for machine learning. It has a comprehensive, flexible ecosystem of tools, libraries and community resources that lets researchers push the state-of-the-art in ML and developers easily build and deploy ML powered applications.

5.1.4 LSTM – Long Short Term Memory

Long-Short-Term Memory Recurrent Neural Network belongs to the family of deep learning algorithms. It is a recurrent network because of the feedback connections in its architecture. It has an advantage over traditional neural networks due to its capability to process the entire sequence of data. Its architecture comprises the cell, input gate, output gate and forget gate.

5.1.5 Flask

Flask (source code) is a Python web framework built with a small core and easy-to-extend philosophy.

5.1.5 HTML

HTML is the standard markup language for Web pages.

5.2 HARDWARE REQUIREMENTS

5.2.1 GPU

The following GPU-enabled devices are supported:

NVIDIA® GPU card with CUDA® architectures 3.5, 3.7, 5.2, 6.0, 6.1, 7.0 and higher than 7.0. See the list of CUDA®-enabled GPU cards.

On systems with NVIDIA® Ampere GPUs (CUDA architecture 8.0) or newer, kernels are JIT-compiled from PTX and TensorFlow can take over 30 minutes to start up. This overhead can be limited to the first start up by increasing the default JIT cache size with: 'export CUDA_CACHE_MAXSIZE=2147483648'

For GPUs with unsupported CUDA® architectures, or to avoid JIT compilation from PTX, or to use different versions of the NVIDIA® libraries, see the Linux build from source guide.

CHAPTER 6

METHODOLOGY

6.1 PREDICTIONS AND ACCURACY

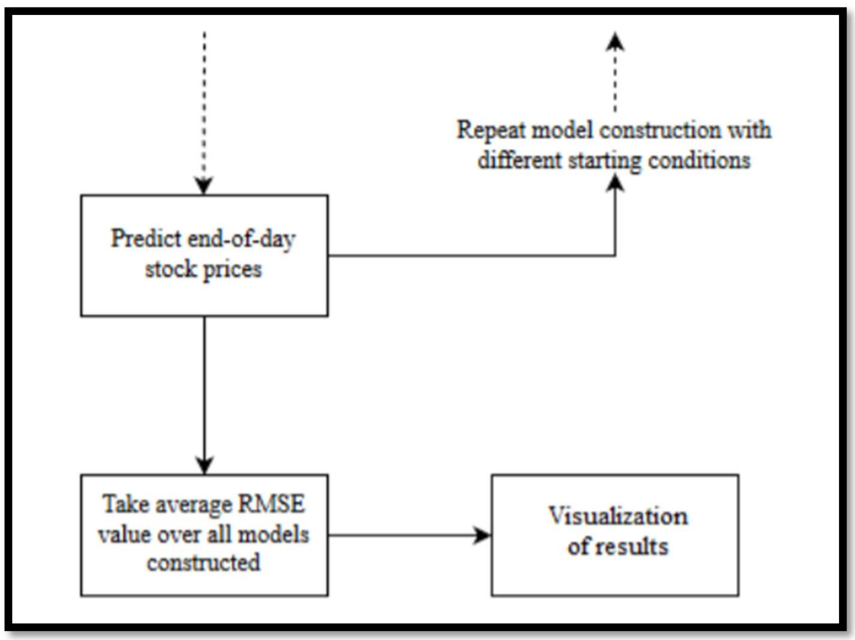

Fig – 6.1: Prediction of end-of-day stock prices

Once the LSTM model is fit to the training data, it can be used to predict the end-of-day stock price of an arbitrary stock. This prediction can be performed in two ways:

1. Static – a simple, less accurate method where the model is fit on all the training data. Each new time step is then predicted one at a time from test data.

2. Dynamic – a complex, more accurate approach where the model is refit for each time step of the test data as new observations are made available.

The accuracy of the prediction model can then be estimated robustly using the RMSE (Root Mean Squared Error) metric. This is due to the fact that neural networks in general (including LSTM) tend to give different results with different starting conditions on the same data. We then repeat the model construction and prediction several times (with different starting conditions) and then take the average

RMSE as an indication of how well our configuration would be expected to perform on unseen real world stock data (figure 6.1). That is, we will compare our predictions with actual trends in stock price movement that can be inferred from historical data.

6.2 VISUALIZATION OF RESULTS

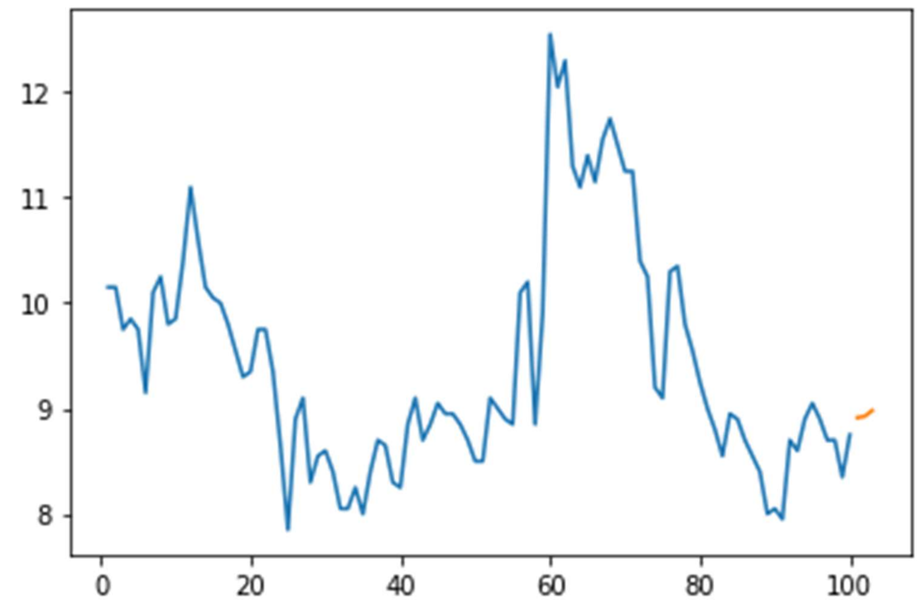

Fig – 6.2: Predicted stock price for IDEA.NS

Figure 6.2 shows the actual and the predicted closing stock price of the company IDEA.NS, a large-sized stock. The model was trained with a batch size of 1130 and 50 epochs, and the predictions made closely matched the actual stock prices, as observed in the graph.

This graph will be finally displayed in a web page.

CHAPTER 7

CONCLUSION AND FUTURE WORK

The results of comparison between Long Short-Term Memory (LSTM) and Artificial Neural Network (ANN) show that LSTM has a better prediction accuracy than ANN. Stock markets are hard to monitor and require plenty of context when trying to interpret the movement and predict prices. In ANN, each hidden node is simply a node with a single activation function, while in LSTM, each node is a memory cell that can store contextual information. As such, LSTMs perform better as they can keep track of the context-specific temporal dependencies between stock prices for a longer period while performing predictions. An analysis of the results also indicates that both models give better accuracy when the size of the dataset increases. With more data, more patterns can be fleshed out by the model, and the weights of the layers can be better adjusted. At its core, the stock market reflects human emotions. Pure number crunching and analysis have their limitations; a possible extension of this stock prediction system would be to augment it with a news feed analysis from social media platforms such as Twitter, where emotions are gauged from the articles. This sentiment analysis can be linked with the LSTM to better train weights and further improve accuracy.

CHAPTER 7

ROLES AND RESPONSIBILITES OF TEAM MEMBERS

NAME	ROLE	RESPONSIBILITY	SKILLS REQUIRED
V Vaishnav	Data Analyst, Backend Programming	Prediction of stock price using LSTM and Server-side connection using Flask (PYTHON)	Machine learning using TensorFlow, Scikit learn, etc. Flask for Web Application
Sidharth R V	Front-end Developer	Developed Templates for different webpages in the web application.	Bootstrap 4, HTML, JavaScript, CSS
Sudarshan U	Front-end Developer	Developed Webpage calling BBC News API creating a small News Aggregator and parsed in the Flask Web Application	HTML, CSS

REFERENCES

[1] Heaton J.B., Polson N.G., Witte J.H. Deep learning for finance: deep portfolios

Applied Stochastic Models in Business and Industry, 33 (1) (2018), pp. 3-12

[2] Gupta, S. and Wang, L.P. (2011) Stock Forecasting with Feed forward Neural Networks and Gradual Data Sub-Sampling. Journal of Intelligent Information Systems, 11.

[3] Zhu, M. and Wang, L.P. (2010) Intelligent Trading Using Support Vector Regression and Multilayer Perceptrons Optimized with Genetic Algorithms. The 2010 International Joint Conference on Neural Networks (IJCNN), Barcelona, 18-23 July 2010, 1-5.

[4] Penglei Gao ,Rui Zhang and Xi Yang (18 August 2020) The Application of Stock Index Price Prediction with Neural Network

[5] Padmaja Dhenuvakonda, R. Anandan, N. Kumar (10.06.2020) STOCK PRICE PREDICTION USING ARTIFICIAL NEURAL NETWORKS

[6] Raghav Nandakumar, Uttamraj K R, Vishal R, Y V Lokeswari(Mar-2018) Stock Price Prediction Using Long Short Term Memory

APPENDIX-I

SAMPLE CODE

```
def predictIdea():

df = pdr.get_data_yahoo('IDEA.NS')

df1 = df. reset_index()['Close']


import numpy as np


from sklearn.preprocessing import MinMaxScaler

scaler=MinMaxScaler(feature_range=(0,1))

df1=scaler.fit_transform(np.array(df1).reshape(-1,1))


training_size=int(len(df1)*0.65)

test_size=len(df1)-training_size

train_data,test_data=df1[0:training_size,:],df1[training_size:len(df1),:1]


import numpy

# convert an array of values into a dataset matrix

def create_dataset(dataset, time_step=1):

    dataX, dataY = [], []

    for i in range(len(dataset)-time_step-1):

            a = dataset[i:(i+time_step), 0]   ###i=0, 0,1,2,3-----99   100
```

```
            dataX.append(a)
            dataY.append(dataset[i + time_step, 0])
    return numpy.array(dataX), numpy.array(dataY)


time_step = 100
X_train, y_train = create_dataset(train_data, time_step)
X_test, ytest = create_dataset(test_data, time_step)


# reshape input to be [samples, time steps, features] which is required for LSTM
X_train =X_train.reshape(X_train.shape[0],X_train.shape[1] , 1)
X_test = X_test.reshape(X_test.shape[0],X_test.shape[1] , 1)


### Create the Stacked LSTM model
import tensorflow as tf
import numpy as np
from tensorflow import keras
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense
from tensorflow.keras.layers import LSTM


model = tf.keras.Sequential([keras.layers.Dense(units=1, input_shape=[1])])
model=Sequential()
model.add(LSTM(50,return_sequences=True,input_shape=(100,1)))
model.add(LSTM(50,return_sequences=True))
model.add(LSTM(50))
```

```
model.add(Dense(1))
model.compile(loss='mean_squared_error',optimizer='adam')

model.fit(X_train,y_train,validation_data=(X_test,ytest),epochs=1,batch_size=64,verbose=1)

train_predict=model.predict(X_train)
test_predict=model.predict(X_test)

##Transformback to original form --- rescaling
train_predict=scaler.inverse_transform(train_predict)
test_predict=scaler.inverse_transform(test_predict)

### Plotting
# shift train predictions for plotting
look_back=100
trainPredictPlot = numpy.empty_like(df1)
trainPredictPlot[:, :] = np.nan
trainPredictPlot[look_back:len(train_predict)+look_back, :] = train_predict
# shift test predictions for plotting
testPredictPlot = numpy.empty_like(df1)
testPredictPlot[:, :] = numpy.nan
testPredictPlot[len(train_predict)+(look_back*2)+1:len(df1)-1, :] = test_predict

x_input = test_data[331:].reshape(1,-1)
```

```
temp_input = list(x_input)
temp_input = temp_input[0].tolist()

# demonstrate prediction for next 10 days
from numpy import array

lst_output=[]
n_steps=100
i=0
while(i<3):

    if(len(temp_input)>100):

        x_input=np.array(temp_input[1:])

        x_input=x_input.reshape(1,-1)
        x_input = x_input.reshape((1, n_steps, 1))

        yhat = model.predict(x_input, verbose=0)
        temp_input.extend(yhat[0].tolist())
        temp_input=temp_input[1:]

        lst_output.extend(yhat.tolist())
        i=i+1
    else:
        x_input = x_input.reshape((1, n_steps,1))
```

```
    yhat = model.predict(x_input, verbose=0)

    temp_input.extend(yhat[0].tolist())

    lst_output.extend(yhat.tolist())
    i=i+1

day_new=np.arange(1,101) #testdata 100indexes
day_pred=np.arange(101,104) #101-131-predicted
plt.plot(day_new,scaler.inverse_transform(df1[1130:]))
plt.plot(day_pred,scaler.inverse_transform(lst_output))
plt.savefig('static/images/IDEA.png')
```

APPENDIX-II

SCREENSHOTS

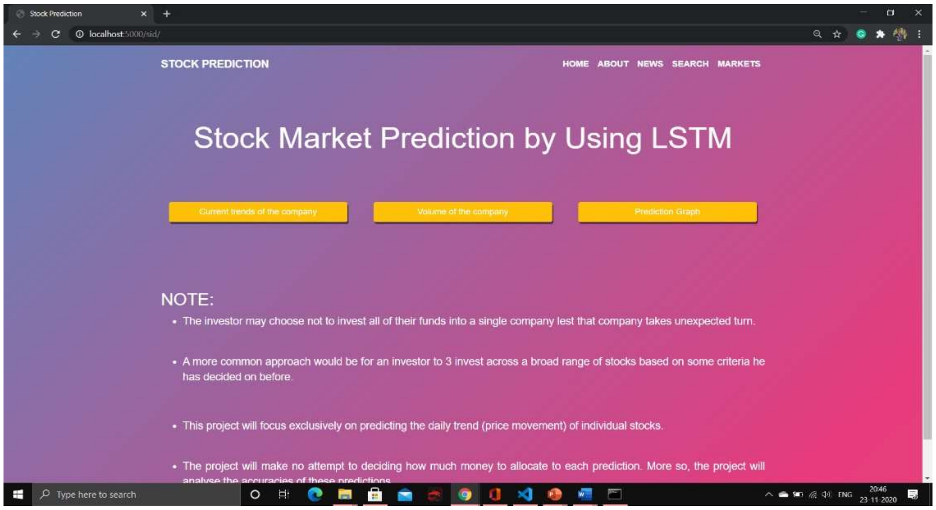

Fig – A.1: Company Dashboard

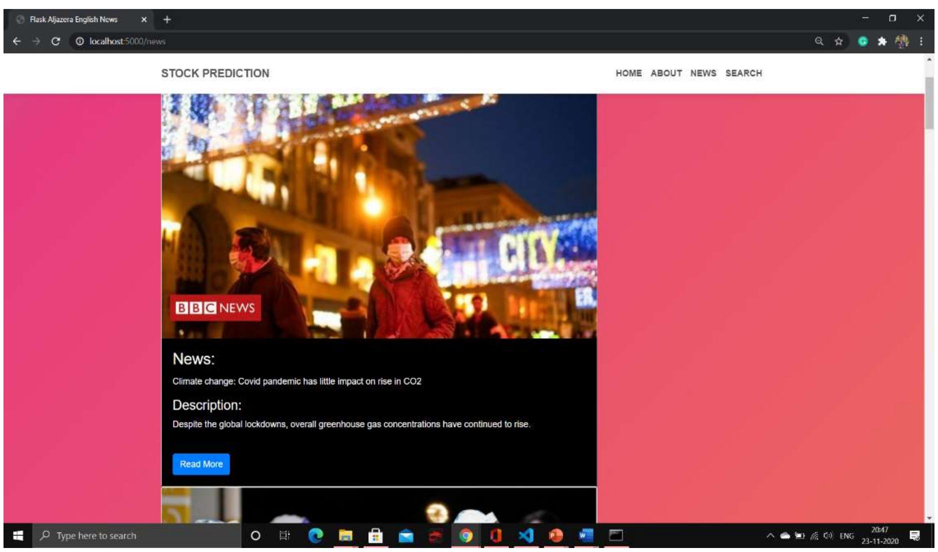

Fig – A.2: News Accumulator Webpage

•

APPENDIX-III

CERTIFICATES


HINDUSTAN
INSTITUTE OF TECHNOLOGY & SCIENCE
(DEEMED TO BE UNIVERSITY)

CERTIFICATE
OF EXCELLENCE

PROJECT EXPO 2020

This certificate is proudly presented to:

Mr. V Vaishnav

has secured **Third Prize** in **Project Expo 2020** Conducted by the Department of Computer Science,School of Computing Sciences at **Hindustan Institute of Technology and Science** on Dec 2nd 2020.

2-12-2020.

Dr.P.Selvi Rajendran
Event-Coordinator

Dr.Angelina Geetha
Head(SCS)


HINDUSTAN
INSTITUTE OF TECHNOLOGY & SCIENCE
(DEEMED TO BE UNIVERSITY)

CERTIFICATE
OF EXCELLENCE

PROJECT EXPO 2020

This certificate is proudly presented to:

Mr. Sidharth R V

has secured **Third Prize** in **Project Expo 2020** Conducted by the Department of Computer Science,School of Computing Sciences at **Hindustan Institute of Technology and Science** on Dec 2nd 2020.

2-12-2020.

Dr.P.Selvi Rajendran
Event-Coordinator

Dr.Angelina Geetha
Head(SCS)